Social Versions System (SVS)

Simon Yuill
6th July 2004

1. outline

Code maintenance is an inherently social act. It involves processes of collaboration, consensus, and conflict resolution, and embodies social processes such as normativisation and differentiation. Software development tools such as CVS (Concurrent Versions System) [1] implicitly formalise such processes and, in doing so, potentially provide means of tracking them.

The aim of this research project will be to develop a graphical tool for displaying such processes as they can be traced in a standard CVS repository. Whilst various tools, such as TkCVS [2], provide graphical interfaces to CVS, these primarily focus on formal aspects of code change. The SVS project will provide a graphical mapping of CVS activity in terms of actions by 'agents' (ie. code contributors) and their development over time. Over the past two years a growing body of sociological analyses have been developed on FLOSS development projects [3]. These have largely made use forms of Social Network Theory and Actor-Network Theory and have primarily looked at the emergent organisational structures evident in the code histories contained in CVS repositories [4]. A few tools have developed in order to support such work. As yet, however, there is no tool specifically designed to provide visualisations and graphical mappings of such analyses and the data they use that is publicly available. Such tools that do exist are only really of benefit to specialists. SVS will seek to develop a more accessible form of graphical representation of the processes under investigation. It will therefore help pull together existing work in this field, make it accessible to a broader audience, and through the provision of a tool, encourage future work.

A unique aspect of SVS will develop through its implementation as part of the spring_alpha project. spring_alpha is a simulated society game in which social change is linked to changes in the fabric of the simulation code. A key part of spring_alpha lies in players being able to re-code aspects of the simulated world as part of the gameplay. Players are thus also developers. The tracking of code development as outlined above will be mapped against the use of code components in game play. In tracking the use of software components alongside development of their code, it will provide a unique insight into the reciprocal nature of a user-developer ethos. Given that the user-developer ethos is one of the key principles, and innovations, of FLOSS this will have significance beyond the spring_alpha project itself.

The development of SVS will be accompanied by a discussion of alternative models of social analysis to those currently employed by existing research in the area, such as Harold Barclay's model of "segmented acephalous networks" [5] and Niklas Luhmann's autopoietic theory [6].

2. existing resources

graphical interfaces and metric tools for CVS

Of the various existing tools for working with CVS the following provide graphical interfaces or measurement utilities relevant to SVS. Some of these will be directly incorporated into the project, others will serve as examples informing its design:

a.) TkCVS is one of the most widely used graphical interfaces available for browsing CVS repositories. The Log interface provides a diagram of different code documents showing their relationship to one another, their authors and events such as branching (variations in code separating into different versions) and merging (combining several variations of the same document into a single version of code) [fig. 1]

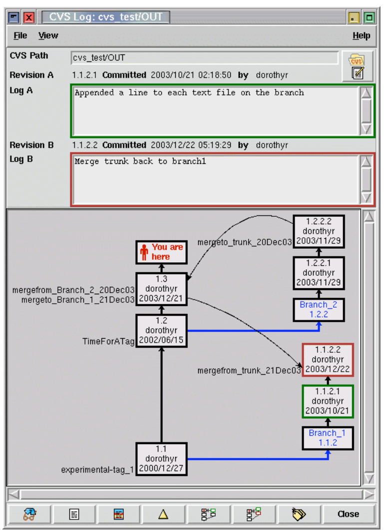

fig. 1 Log interface for TkCVS

Another type of interface shows a colour-coded representation of a section of code, along with information about the authors, and the document versions where particular lines were last ammended [fig. 2]. This is one of the most common forms of CVS display, the colours being used to highlight particular authors, or differences in different versions of the same code. Fig .2b shows a comparable display from the CodeStriker tool [7].

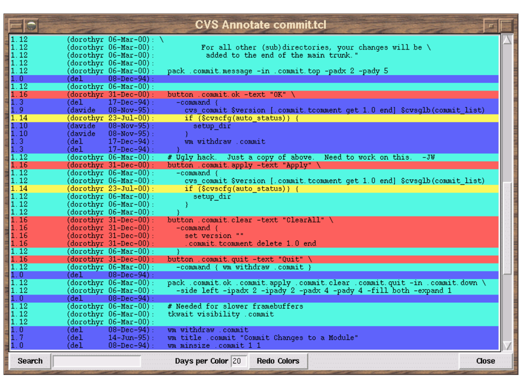

fig. 2a. colour annotated code, TkCVS

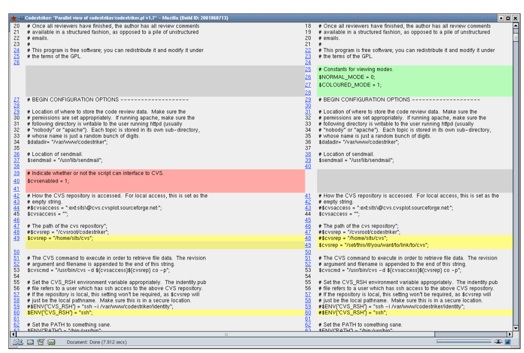

fig. 2b. code comparison, CodeStriker

b.) StatCVS [8], CVSPlot [9] and CVSMonitor [10] are a range of tools which provide various statistical measurements of activity within a CVS repository, such as the lines of code added over time [fig. 3].

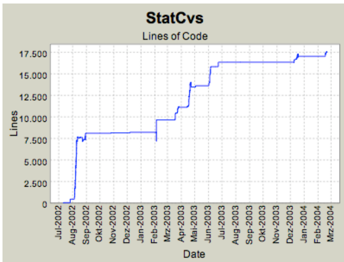

fig. 3a. StatCVS

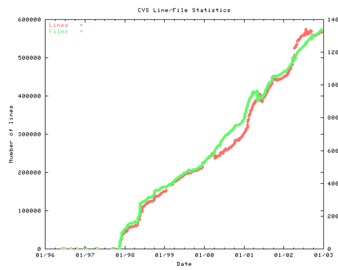

fig. 3b. CVSPlot

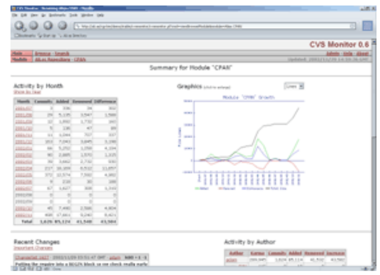

fig. 3c. CVSMonitor

c.) Code Historian [11] is another tool suite which provides a range of displays of activity within a CVS repository over time as well as tracking the contributions of different contributors down to such detail as number of bytes contributed. The tool is primarily a productivity tool, however, geared towards the needs of project managers keeping track of how much work developers are doing [fig. 4]:

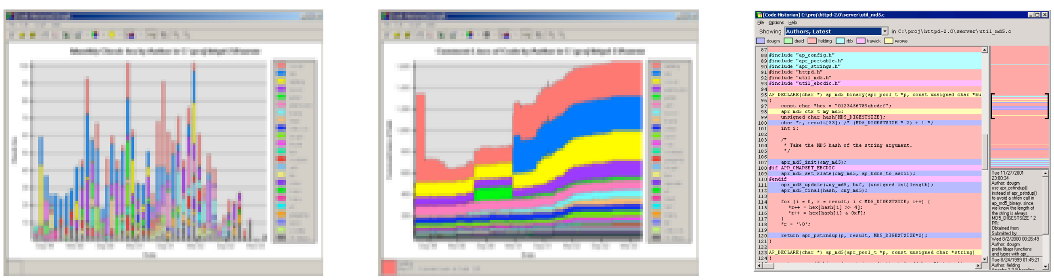

fig. 4. various interfaces from Code Historian

code documentation, analysis and performance metric tools

Code documentation tools are primarily used to generate information that has been written alongside code using either comments or special mark-up languages. These provide notes for developers on how the code may be used and expanded. Certain documentation tools, such as Doxygen [12], can also generate diagrams showing the relationships between components in a program (such as different classes in an Object Orientated language like C++). Debugger and performance profiling tools, such as DBX [13], GDB [14] and gprof [15], can also be used to generate information about code whilst it is running, such as which parts of a program are being used the most, how they interact with one another, and how long they take to perform their task. This information can be converted into diagrams that show how parts of a program relate to one another during usage [fig. 5].

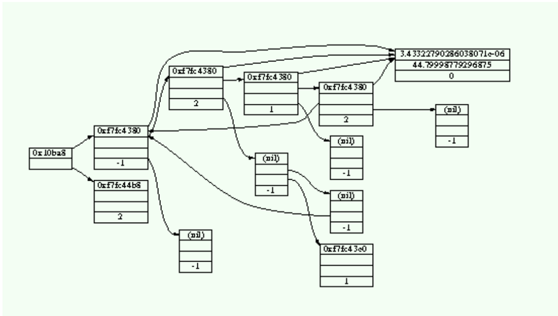

fig. 5a. data structure diagram

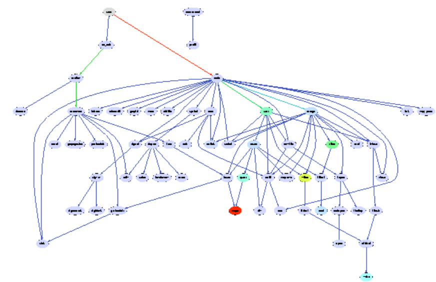

fig. 5b. function call diagram

sociological studies of collaborative programming projects

Some of the existing sociological studies of CVS repositories have created tools to aide gathering information and visualising it. One of the most developed of these is CODD [16]. This generates statistical information about the activities of authors wiothin a CVS repository relevant to a sociological study. This data is then visualised through existing graphing tools. Nicolas Ducheneaut developed his own tool that can access a CVS repository [17], along with the mailing lists that accompany it, and correlate interactions between authors within the repository to threads of discussion in the mailing list [fig. 6].

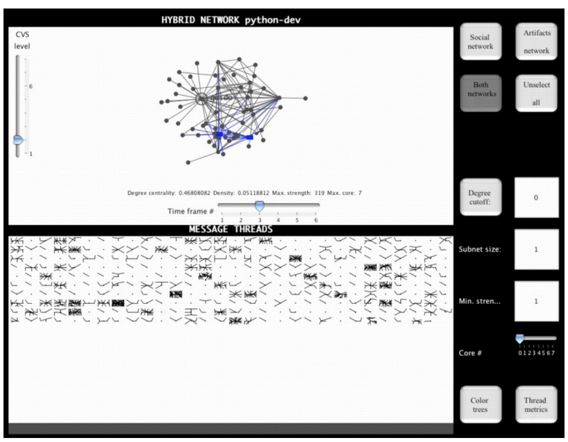

fig. 6. diagram correlating relations between developers to mailing list discussion

3. design issues

One of the key features of SVS will be to map developments within a given project over time and enable commentary and analysis to be combined with that. This would include such aspects as changes to code as well as interactions of the code producers with that. This therefore suggests a visual structure expressing different linear strands and moments of development related to a timeline. The interface would allow different forms of information and analysis to be selected and displayed. This follows on the research of Edward Tufte [18] who has shown that graphical timeline 'narrativisations' of data are often the best way of making developments in sets of complex information explicit and accessible to the viewer [fig. 7]. In these diagrams a variety of visual forms are combined to express multiple aspects of the data displayed.

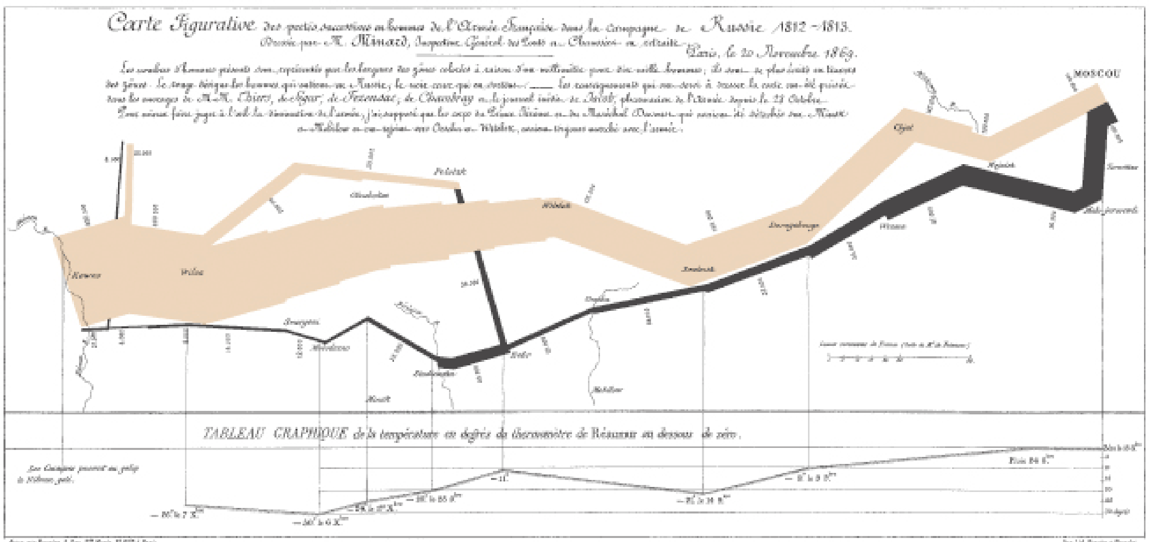

fig. 7a. diagram of Napoleon's march to Moscow

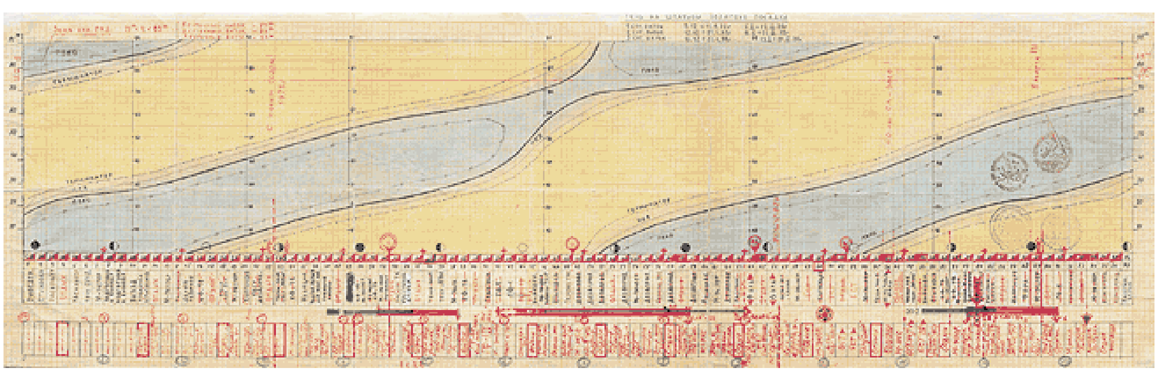

fig. 7b. cosmonaut journal from Salyut 6 mission

Whilst some of the CVS tools provide time-based data, this is almost exclusively a record of quantities of code rather than events within a project, such as branching paths of development, or interactions between developers. With the exception of Ducheneaut's system, most visualisations of CVS projects that provide information about such interactions are presented as static snapshots, or a series of snapshots over time. SVS will focus on a hybrid visualisation

combining timeline 'narrativisation' along with more detailed views of particular data-sets. These details will include, for example, groupings of developers or code fragments, graphical indicators of code change, and relations of code components within the active software.

Fig. 8 shows some possible forms of visualisation to be explored further in SVS. Fig. 8a shows multiple timelines of different processes occurring simultaneously - software usage and development - enabling them to be cross-referenced. Figs. 8b and 8c shows two variations on more detailed timeline forms showing code branching and merging. In 8c line thickness is used as an indicator of growth in code size, but could also show factors such as usage or other activity. These could be combined with colour coding or overlaid graphs showing developers' interactions, as well as notational and textual commentary such as code documentations notes and extracts from mailing list discussions. This may, in some ways, be reminiscent of the graphical cosmonaut journal in fig. 7b.

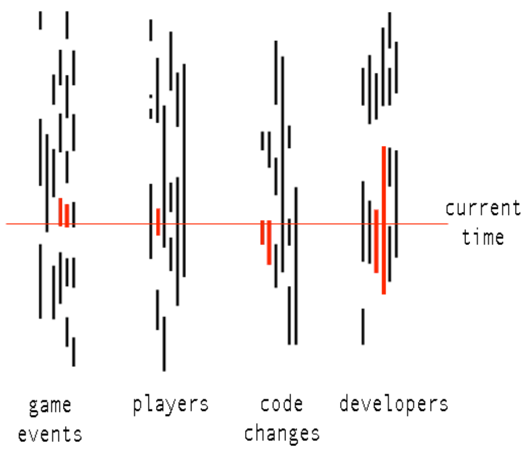

fig. 8a. comparative timelines

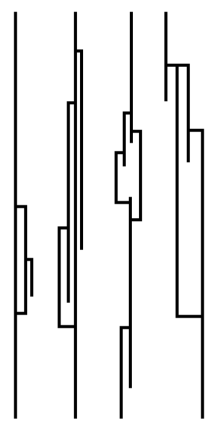

fig. 8b. branch and merge patterns

fig. 8c. branch, merge and code growth

Referencing between live gameplay and SVS is shown in fig. 9. A menu of available game actions for an object is linked to diagrams displaying the object's relation to the actions of other objects and also the game actors interacting with it. These could also be referenced to code development data showing the relationship between modifications to game code and events in the spring_alpha game.

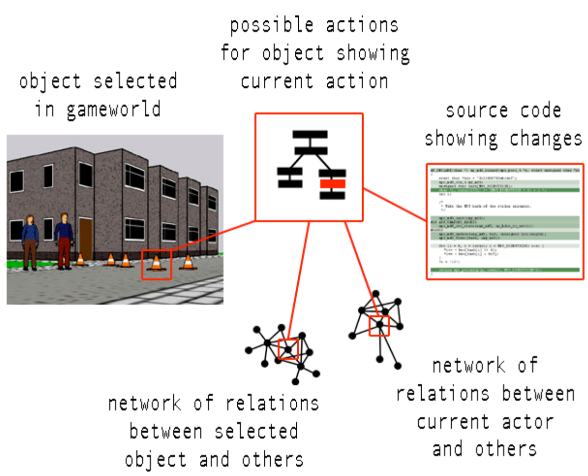

fig. 9. cross-referencing between gameplay and code analysis

Other diagrams could be forms of clustering and networks of relations between developers (as in some of the studies mentioned above) as well as components of the software and events within gameplay. By allowing a variety of visualisation styles to be combined, quite rich information pictures can be created of the activities underlying both code development and usage.

As it is being incorporated into a gaming system, the interface for the spring_alpha implementation will adopt some of the display mechanisms from games. The visualisation will be explorable like a terrain with the possibility to zoom out for an overview and zoom in for greater detail. Views from "further away" will be simplified to show more general data, while zooming in will reveal more detailed information (this can be enabled through the built-in Level Of Detail functionality of game rendering engines). In an exhibition presentation of spring_alpha the normal gameplay view and SVS analysis will be shown side-by-side but on different projections or screens. A common highlighting mechanism (such as colour-coded outlines) will be used to display correlations between items in each kind of display.

references

1. CVS, https://www.cvshome.org

2. TkCVS, http://www.twobarleycorns.net/tkcvs.html

3. Geert Lovink recently collated a list of some of these, some of which are discussed in this proposal (the listing was published on the Oekonux mailing list):

   Crowston, K., & Howison, J. "The social structure of Open Source Software development teams." Submitted for presentation at the "2003 International Conference on Information Systems." (2003) http://crowston.syr.edu/papers/icis2003sna.pdf

   Ducheneaut, N., "The reproduction of open source software communities." Ph.D. work at Berkeley. (not dated) http://www2.parc.com/csl/members/nicolas/open-source.html

   Gao, Y., "Topology and evolution of the Open Source Software community." MS Thesis. University of Notre Dame. (2003), http://www.nd.edu/~oss/Papers/ygaoThesisFinal_12-2-03.pdf

   Gao, Y., Madey, G., & Freeh, V., "Modeling and simulation of a complex social system: A case study." (2003) http://www.nd.edu/~oss/Papers/SimulationSym37_gao.pdf

   Kuwabara, K., "Linux: A bazaar at the edge of chaos." First Monday, vol. 5, no. 3 (March 2000). http://www.firstmonday.org/issues/issue5_3/kuwabara/index.html

   Lopez-Fernandez, L., Robles-Martinez, G., & Gonzalez-Barahona, J.M., "Applying social network analysis to the information in CVS repositories." (2004), http://libresoft.dat.escet.urjc.es/html/downloads/sna-icse-2004.pdf, http://msr.uwaterloo.ca/slides/LopezFernandez.pdf

   Madey, G., Freeh, V., & Tynan, R. "Modeling the F/OSS Community: A quantitative investigation." In S. Koch (ed.), Free/Open Source Software Development. Idea Publishing. (2004), http://www.nd.edu/~oss/Papers/BookChapter.pdf

   Meyer, P.B., "Episodes of collective invention." Working Paper 368. U.S. Department of Labor, Bureau of Labor Statistics. (2003), http://opensource.mit.edu/papers/meyer.pdf

   Mockus, A., Fielding, R.T., & Herbsleb, J.D., "Two case studies of Open Source Software development: Apache and Mozilla." (not dated), http://www.research.avayalabs.com/techreport/ALR-2002-003-paper.pdf

   Myers, C.R., "Software systems as complex networks: Structure, function, and evolvability of software collaboration graphs." Cond-mat/0305575. (2003), http://arxiv.org/abs/cond-mat/0305575

   OMahony, S., & Ferraro, F., "Managing the boundary of an open project." Prepared for the Santa Fe Institute (SFI) Workshop on .The Network Construction of Markets. (2003), http://opensource.mit.edu/papers/omahonyferraro.pdf

   te Meerman, S., "Puzzling with a top-down blueprint and a bottom-up network: An explorative analysis of the Open Source world using ITIL and Social Network Analysis." MS Thesis, University of Groningen, the Netherlands. (2003), http://opensource.mit.edu/papers/meerman2.pdf

   Waguespack, D.M., & Fleming, L., "Penguins, camels, and other birds of a feather: The emergence of leaders in open innovation communities." (2004), http://faculty.haas.berkeley.edu/elf/ECON222/Waguespack%20Fleming.pdf

   Wasko, M.M., & Teigland, R., "The provision of online public goods: Examining social structure in an electronic network of practice." 23rd International Conference on Information Systems. (2002), http://www.cs.vu.nl/~aschwarz/workshop/abstracts/teiglanda.htm

   Xu, J., & Madey, G., "Exploration of the Open Source Software community." NAACSOS Conference 2004. (2004), http://casos.isri.cmu.edu/events/conferences/2004_proceedings/XuJin1.pdf

   see also:

   Rishab Aiyer Ghosh, "Clustering and dependencies in free/open source software development: Methodology and tools," First Monday, volume 8, number 4 (April 2003), http://www.firstmonday.dk/issues/issue8_4/ghosh/index.html

4. Social Network Theory is a method of mapping and analysing human organisations or groups in terms of the interactions between their members, key developers include Jeremy Boissevain and Mark Granovetter. Actor-Network Theory is a form of analysis of the relationships between human and non-human components of complex social systems, originally developed by Michel Callon and Bruno Latour.

5. Harold Barclay "Segmental Acephalous Network Systems: alternatives to centralised bureacracy," The Raven (7, 1989)

6. Niklas Luhmann "Social Systems," Stanford University Press (Stanford 1995), "Art as a Social System," Stanford University Press (Stanford 2000)

7. Codestriker, http://codestriker.sourceforge.net

8. StatCVS, http://statcvs.sourceforge.net

9. CVSPlot, http://cvsplot.sourceforge.net

10. CVSMonitor, http://ali.as/devel/cvsmonitor

11. Code Historian, http://codehistorian.com

12. Doxygen, http://www.stack.nl/~dimitri/doxygen

13. DBX, http://www.physics.utah.edu/~p573/hamlet/lessons/dbx/dbx/dbx.html

14. GDB, http://www.gnu.org/software/gdb/gdb.html

15. gprof, http://www.cs.utah.edu/dept/old/texinfo/as/gprof.html

16. CODD, http://orbiten.org/codd

17. Nicolas Ducheneaut, "The Reproduction of Open Source Software Communities," (not dated) http://www2.parc.com/csl/members/nicolas, unfortunately the software is no longer available.

18. Edward Tufte, "The Visual Display of Quantative Information," Cheshire, Connecticut: Graphics Press (1983), "Envisioning Information," Cheshire, Connecticut: Graphics Press (1990), "Visual Explanations: Images and Quantities, Evidence and Narrative," Cheshire, Connecticut: Graphics Press (1997)